

Compiler Design in C - Chapter 7 (Optimization Strategies)

The authors of this work would like to present the basic types of optimizations and show some optimization techniques. At first, we divide and describe three categories of optimizations as follows: Parser Optimizations, Linear (Peephole) Optimizations and Structural Optimizations.

Parser Optimizations includes all optimizations that can be done by the parser itself. The purpose of this part is to generate proper code by using following methods: using logical l-values rather than physical values, minimizing the number of goto branches etc. We mainly focus on *Intrinsic functions*.

Linear (Peephole) Optimizations covers various optimizations that cannot be done in the parser, but which can be done by an auxiliary optimizer pass that goes through the compiler's out - put in a linear fashion, from top to bottom. We explain *Strength Reduction* by a simple example on multiplication that is replaced by a left shift. Afterwards, there are shown two examples. The first one demonstrates when the usage of this technique makes sense and its usage is an advantage. In the second example the usage should be determined by the optimizer as there the code-size-versus-execution-speed trade-off problem needs to be solved. Also we briefly cover *Constant Folding and Constant Propagation, Dead Variables and Dead Code Elimination* as these techniques have already taken part during the lectures. Finally, we mention *Hardware problems* and usage of the ANSI C *volatile* keyword to prevent these problems.

At the end, we cover the principle of Structural Optimizations by mentioning two difficulties that come with it and we explain Postfix or Reverse Polish Notation (RPN) that is used here. There are described several techniques as follows: *Common-Subexpression Elimination, Register Allocation* and, *Replacing Indexes with Pointers and Loop-Invariant Code Motion*.

The contribution of this presentation is to give to a listener the basic ideas of Optimization Strategies that are used in compilers. The final presentation can server as an overview of Optimization Strategies and techniques.